Wirtschaftsinformatik 2021 Proceedings

Track 10: Design, management and impact of AI-based systems

# Leveraging Text Classification by Co-training with Bidirectional Language Models – A Novel Hybrid Approach and its Application for a German Bank

Roland Graef
*Ulm University, Germany*

# Leveraging Text Classification by Co-training with Bidirectional Language Models – A Novel Hybrid Approach and its Application for a German Bank

Roland Graef[1]

[1] University of Ulm, Institute of Business Analytics, Ulm, Germany
roland.graef@uni-ulm.de

**Abstract.** Labeling training data constitutes the largest bottleneck for machine learning projects. In particular, text classification via machine learning is widely applied and investigated. Hence, companies have to label a decent amount of texts manually in order to build appropriate text classifiers. Obviously, labeling texts manually is associated with time and expenses. Against this background, research started to develop approaches exploiting the knowledge contained in unlabeled texts by learning sophisticated text representations or labeling some of the texts in an automated manner. However, there is still a lack of integrated approaches, considering both types of approaches to further reduce time and expenses for labeling texts. To address this problem, we propose a new hybrid text classification approach combining recent text representations and automated labeling approaches in an integrated perspective. We demonstrate and evaluate our approach using the case of a German bank where the approach could be applied successfully.

**Keywords:** Machine Learning, Text Classification, Co-training, Bidirectional Long Short-Term Memory Networks

## 1 Introduction

Machine learning is becoming a main driver for automating processes and developing new business models as well as products [1]. As a consequence, companies worldwide and of all sizes are increasingly investing in machine learning [2]. For instance, machine learning is also becoming more and more established amongst German companies as a recent study by the International Data Group [1] shows. In this regard, the share of German companies dealing with the application of machine learning has risen by 20% up to 73% compared to the year 2019. Since machine learning can particularly be used to find sophisticated patterns in texts, it excels at the task of text classification [3–10]. Indeed, organizations use machine learning for text classification within diverse value creating tasks. PayPal, for instance, as an operator of a worldwide online payment system successfully employs the machine learning platform RapidMiner for a real-time text classification of customers' feedback messages in terms of sentiment. Thereby, PayPal aims to enable an instant reaction to displeased

16th International Conference on Wirtschaftsinformatik,
March 2021, Essen, Germany

customers for preventing churn [11]. In order to develop a machine learning application for text classification a decent amount of training data in terms of labeled texts is required. In practice, organizations prefer, or may even be forced, to collect training data by labeling their internal texts so that machine learning approaches can learn the specific context. In particular, if a company has a domain-specific language or specific classes are required, it is necessary to label internal texts. For example, companies in specific domain areas (e.g. insurance) developing a text classification approach for an inbound routing of incoming customer mails, need to use their customers' texts and label them by hand with respect to their predefined desired domain-specific classes. Actually, labeling training data increasingly represents the largest bottleneck for machine learning projects [12]. A recent study found that 25% of time for machine learning projects is allocated to data labeling [13]. Consequently, labeling large amounts of texts as training data for building an adequate text classification approach via machine learning represents a time consuming and expensive task [3–6, 12, 13]. These expenses are even further increased if domain experts are required to label texts.

To address this challenge and tap the potential of text classification, research has started to develop approaches exploiting unlabeled texts in order to enhance text classifiers trained only on a small set of labeled texts [3–8, 14–19]. On the one hand, authors focus on a sophisticated semantic text representation by training deep learning models on a large amount of unlabeled texts [7, 8, 14–18, 20]. By this means, downstream text classifiers based on machine learning are supported in learning to adequately distinguish classes. On the other hand, literature provides approaches to increase the amount of labeled data based on automated labeling procedures [3–6, 19]. However, there is still a lack of integrated approaches considering both. Therefore, in the problem context of reducing time and expenses associated with manually labeling texts for text classification approaches based on machine learning, merging these two research streams seems very promising to cope with our problem. To address this research gap, we propose a new hybrid text classification approach leveraging the capabilities of text classifiers based on recent text representations as well as automated labeling approaches by exploiting unlabeled data in an integrated approach. Thereby, we aim at reducing time and effort for labeling texts as well as enhancing text classification accuracy when the number of labeled texts is limited.

Following a design-oriented approach (cf., e.g. Peffers et al. [21]), the remainder of this paper is structured as follows: In the next section, we provide an overview of the related work and the research gap. In Section 3, we propose a hybrid text classification approach combining recent text representation and automated labeling approaches. In Section 4, we demonstrate and evaluate our approach based on the case of a German direct banking institution. Finally, we conclude with a summary of the findings, a discussion of limitations and an outlook on future research.

## 2 Related Work and Research Gap

Text classification via machine learning approaches is widely applied and investigated by recent research [3–10]. Since labeling a large amount of texts as training data for

machine learning is a time consuming and expensive task, particularly if domain experts are required [3–6, 12], literature started to develop approaches which require a rather small amount of manually labeled texts and therefore exploit unlabeled texts. A recent survey examines a wide range of these so-called semi-supervised approaches while presenting a respective taxonomy [22]. In case of text data, however, research particularly focuses on producing a sophisticated semantic representation of text [7, 8, 14–18, 20] or develops approaches to label data in an automated manner [3–6, 19]. To cope with our problem of further reducing time and expenses required for labeling texts, both research streams seem promising.

Recent surveys already review the great evolvement of semantic representations of text in order to solve diverse downstream tasks of Natural Language Processing as, for example, text classification [9, 10]. In this regard, research aims at deriving so-called embeddings, representing the semantics of texts within dense vectors. In this context, embeddings are usually gained by training Neural Networks on large text corpora so that the Neural Networks learn to decode the semantic meaning of words based on the context [7, 8, 14–18, 20]. Although embeddings can be trained simultaneously when training Neural Networks for text classification (e.g. by using an embedding layer [23]), such an approach does not profit from unlabeled texts [10]. As a consequence, research started to develop Neural Network approaches, which can be trained on unlabeled texts. Subsequent to training, these Neural Networks can be applied to produce embeddings as an input for downstream text classifiers.

Embeddings can be particularly divided into single global [14, 15, 20] and context-dependent representations of words [7, 8, 18, 24, 25]. Research focusing on embeddings started with the development of single global representations of words and gained high popularity with approaches as, for instance, Word2Vec [14] or GloVe [15]. As both of these famous examples are limited to the vocabulary they have been trained on, more general approaches have been investigated. For example, the widely used single global embeddings from Kim et al. [20] overcome the issue of building embeddings for unknown words by processing words character by character. However, due to their global representation of context, single global embeddings fail at accurately representing polysemous words (e.g. the word “apple” may refer either to the fruit or the company) [9, 10]. Consequently, research addressed this challenge by building context-dependent representations using single global embeddings as a basis [7, 8]. To do so, authors started to exploit the capabilities of bidirectional Long Short-Term Memory Networks (biLSTMs) [7, 8, 18, 24, 25]. BiLSTMs constitute a specific type of Recurrent Neural Networks, which are designed to process sequential data from both sides while learning long term dependencies as, for instance, contextual information contained in natural language [18, 26]. In terms of text, mostly words of a sentence are passed step-by-step to the biLSTMs as sequential data. For training biLSTMs with unlabeled data usually the same texts in two different languages [8, 17] or words from the surrounding context [7, 18] are used as labels. As a result, a biLSTM can be used to produce the context-dependent embedding of a word given its preceding and following words in a sentence [7, 9]. One of the first authors approaching context-dependent word embeddings with biLSTMs are Kawakami and Dyer [17], who trained their biLSTM by using cross-lingual supervision. More precisely, their approach

predicts a target word from a different language based on a sentence from the source language. By this means, the representation of polysemous words is context-dependent as these words are usually not polysemous in the target language (e.g. translating the word of the fruit “apple” to the German word “Apfel” avoids any confusion with the company when context-dependent embeddings are learnt). Further authors using training data from different languages, developed the well-known context-dependent word embeddings CoVe on the basis of English-to-German translation [8]. On this account, they exploited an encoder and decoder architecture, trained with GloVe [15] as input vectors for the source language. Subsequent to training, the encoder, consisting of two layers of biLSTMs, is used to produce the CoVe embeddings. Others obtained their context-dependent embeddings by training their biLSTM models to predict a target word based on the preceding and following sequences of words in sentences [7, 18]. Melamud et al. [18], for instance, further developed the idea of Word2Vec [14] by additionally creating context vectors to single global word embeddings with their Context2Vec approach. Thereby, they demonstrated that their context vectors outperform averaged Word2Vec embeddings to represent the context of sentences in different Natural Language Processing tasks. Meanwhile, the probably most popular context-dependent embeddings based on biLSTMs are those of the ELMo approach [7]. The concept behind ELMo is to train $L$ layers of biLSTMs for predicting a target word based on its surrounding context words while also using single global embeddings as input. In contrast to other approaches, ELMo not only uses the top layer biLSTM to produce the context-dependent embeddings but rather collapses the output of all layers based on a task specific weighting to gain the ELMo embeddings. By this means, the single global embedding and multiple context-dependent embeddings are combined and offset with each other.

Other researchers exploit unlabeled text data by developing approaches to automatically label data, which subsequently can be used as training data [3–6]. To do so, these authors rely on the famous co-training approach [19]. The idea behind co-training is to train two classifiers on the same training data, but provide each with a different view (e.g. set of features) of the data. Subsequently, each classifier can be applied to classify some of the unlabeled data, which can in turn be used to train the respective other classifier. This procedure can be repeated until a stopping condition is met (e.g. if all unlabeled data has been labeled or a given number of iterations is reached). Authors benefitting from co-training within their text classification approaches either use different representations of text as views [5, 6], adjust the co-training approach [3, 27] or even do both [4]. For instance, Kim et al. [6], employed three different representations of text as views and trained a classifier for each view within their co-training approach. More precisely, their respective views of text representations constitute the statistical term weighting representation tf-idf, the generative topic model Latent Dirichlet Allocation as well as the Doc2Vec approach, which is an evolution of the Word2Vec approach for whole documents. Others as, for example, Katz et al. [3], overthought the co-training approach by saving and using all classifiers within each iteration of the co-training process to classify texts. As a result, the most recent classifiers are used as an ensemble to classify the test data.

To sum up, both, context-dependent embeddings based on biLSTMs as well as co-training seem very promising means to reduce time and expenses associated with manually labeling texts as training data by exploiting the knowledge contained in unlabeled texts. However, first promising approaches dealing with both types of knowledge expansion through unlabeled texts do not fully exploit the capabilities of individual context-dependent embeddings based on biLSTMs [4, 28] or require additional human knowledge for modeling [5]. Chen et al. [4], employ co-training by using a single global embedding in terms of Word2Vec as one view of the text and context-dependent embeddings in terms of ELMo as the other view. Therefore, one of their classifiers is only provided with single global embeddings of text and cannot resolve context-dependent relationships as, for instance, polysemous words. Lim et al. [28], employ co-training from a broader view by combining multiple context-dependent embeddings from different biLSTMs. Thus, they only add up information contained in multiple embeddings using them as different views in co-training but do not take advantage of the information offered by already one context-dependent embedding based on biLSTMs. Actually, a more in-depth combination of context-dependent embeddings based on biLSTMs and co-training could be integrated into their approach to reach further improvement. Karisani et al. [5] who rely on context-dependent embeddings based on the very recent bidirectional transformer architecture [9], require further human knowledge to model different concepts of texts, which are subsequently used as different views in co-training. Hence, time and expenses for manual tasks are not necessarily reduced. Indeed, further exploiting the information contained in embeddings based on the transformer architecture via co-training is hardly possible without human modeling of features. Although the transformer architecture is entitled bidirectional, it is rather alldirectional in the sense that it processes sentences from both directions at the same time instead of once in each direction [16]. Hence, context-dependent embeddings based on the transformer architecture do not provide two different views of the same text, which is the prerequisite for designing co-training approaches.

To the best of our knowledge, so far none of the studies in text classification has considered embeddings based on biLSTMs in conjunction with co-training while at the same time taking an integrated perspective by not only using embeddings as different views in co-training but rather combining research streams by merging co-training into context-dependent embeddings based on biLSTMs. To address this gap, we follow a design oriented approach (cf., e.g. Peffers et al. [21]) and aim at developing, a novel hybrid text classification approach, combining embeddings based on biLSTMs with co-training in a well-founded way.

# 3 Hybrid Approach for Leveraging Text Classification by Co-training with Bidirectional Language Models

## 3.1 Basic Idea and Overview of the Hybrid Approach

The aim of this paper is to develop a text classification approach, which reduces the amount of labeled texts required to train sound machine learning classifiers and consequently reduces time and expenses to label texts by hand. To reach this goal our approach exploits unlabeled data, on the one hand, for generating context-dependent embeddings based on biLSTMs for a sophisticated text representation. On the other hand, we enable an automated labeling of texts to expand training data by relying on the well-known co-training approach. By these means, our approach is well-suited to leverage the capabilities of text classification approaches when only a small amount of labeled data is available. Hence, accuracy of text classification can be improved while simultaneously reducing time and expenses for labeling texts by hand. Our approach comprises two phases (cf. Figure 1).

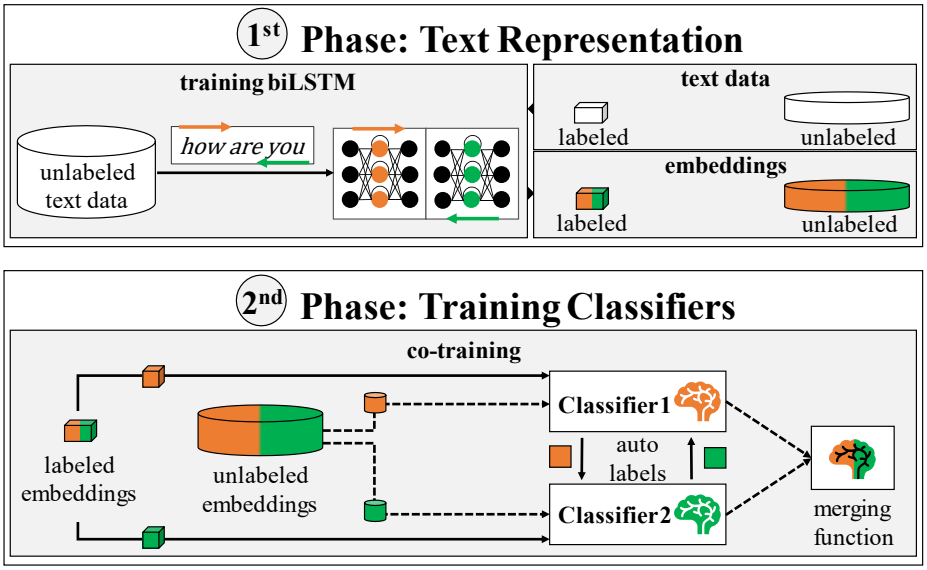


**Figure 1.** Hybrid text classification approach

In the first phase, the semantic information contained in text is decoded into machine readable form. To do so, we rely on biLSTMs while suggesting approaches from language modeling for training biLSTMs on a large corpus of unlabeled texts. Afterwards, the trained biLSTMs can be used to represent text data in terms of context-dependent embeddings. Since biLSTMs decode text once in reading direction and once in the opposite direction, they provide context-dependent embeddings with two different views by design thereby drawing on the past and future context of the text,

respectively. The second phase of our approach builds upon the context-dependent embeddings of the first phase. As context-dependent embeddings based on biLSTMs provide two different views on the text by design, we make use of these views by designing a co-training approach. Accordingly, two classifiers are trained, each on a different view, to label text data for the respective other classifier in an automated manner. We refer to these labels as auto labels. Obviously, a larger training set enhances the text classification capabilities of the classifiers. At last, we propose a merging function to combine the results of the trained classifiers for deployment. In the following subsections, we present our hybrid approach for reducing time and expenses associated with manual text labeling in detail.

### 3.2 Text Representation: Context-Dependent Embeddings Based on BiLSTMs

The aim of the first phase is to reach a context-dependent representation of text based on unlabeled text data to make the information contained in natural language accessible for classifiers. Additionally, the desired representation shall offer two different views on the text so that the following phase can profit of these views in terms of co-training. Since context-dependent embeddings based on biLSTMs are the most recent and popular text representations offering two different views by design [9], we rely on approaches based on biLSTMs. On this account, research in context-dependent embeddings heavily relies on the concept of language modeling while reaching convincing results [4, 7, 9, 10, 18, 24, 25, 28]. Consequently, we propose to train biLSTMs based on language modeling. Note that although we suggest to use language modeling to build context-dependent embeddings, our approach is not limited to text representations based on language modeling approaches. Indeed, our approach can exploit each context-dependent embedding based on biLSTMs. For instance, CoVe embeddings, using an encoder and decoder architecture on the basis of English-to-German translation [8], are also very well-suited for usage in our approach.

Language modeling can be described as estimating a probability distribution over a sequence of words based on the preceding or following words [7, 9, 10, 25]. More precisely, for a given sequence of $n$ words $(w_1, \dots, w_n)$ language modeling aims to estimate the probability of that sequence by factorizing it either based on the past or future context. In case of the past context, the factorization is described by the following equation:

$$p(w_1, \dots, w_n) = \prod_{i=1}^{n} p(w_i | w_1, \dots, w_{i-1}) \quad (1)$$

To estimate the conditional probability $p(w_i | w_1, \dots, w_{i-1})$ unidirectional Long Short-Term Memory Networks (LSTMs) are trained to predict the word $w_i$ given its past context words $(w_1, \dots, w_{i-1})$. By this means, unlabeled text data can be exploited for training. As a by-product, LSTMs learn to represent internally the context of the target word $w_i$, which can be extracted as the context-dependent embedding. In order to do so, LSTMs keep an internal memory, combining knowledge of previously processed words with the words they are currently processing. Since the internal memory is capable of storing information over an extraordinary long period, even long-term dependencies among words are established. In the same vein, language modeling can

be approached by relying on the future context of words, as shown in the following equation:

$$p(w_1, \ldots, w_n) = \prod_{i=1}^{n} p(w_i | w_{i+1}, \ldots, w_n) \quad (2)$$

Similarly, unidirectional LSTMs are trained to predict the word $w_i$, but this time based on the future context comprising the words $(w_{i+1}, \ldots, w_n)$. Employing both, LSTMs based on the past context as well as LSTMs based on the future context of words, results in biLSTMs. In this regard, embeddings from the past and future context are concatenated, resulting in the context-dependent embeddings based on biLSTM. Further on, popular research [7, 8, 25] suggests to represent the past $(w_1, \ldots, w_{i-1})$ and future $(w_{i+1}, \ldots, w_n)$ context words by means of single global embeddings (e.g. GloVe [15]) as input for the biLSTMs to improve learning the context-dependent representation. A further enhancement is proposed by the most recent ELMo approach [7] by training *L* layers of biLSTMs, each further processing the output of the preceding layer. On this basis, each biLSTM layer outputs a context-dependent embedding. Finally, the ELMo embedding is determined by offsetting all context-dependent embeddings as well as the single global embedding, serving as input, with each other based on a task specific weighting. In fact, training biLSTMs to reach context-dependent embeddings requires a large amount of unlabeled text data and computational resources. The ELMo approach, for example, was trained on the basis of the one billion word benchmark [7]. Hence, it is common practice, to make use of a pretrained biLSTM approach to generate context-dependent embeddings. Subsequently, the trained biLSTM can be used to generate context-dependent embeddings representing the unlabeled as well as labeled text data necessary for training text classifiers in the following phase. If the amount of unlabeled text data is large enough, both phases of our approach can rely on the same unlabeled texts.

To sum up, state of the art approaches train biLSTMs on an huge amount of unlabeled text data via language modeling. By this means, biLSTMs learn to provide a text representation in terms of context-dependent embeddings based on two views. On this basis, both labeled and unlabeled text data are represented by means of context-dependent embeddings. The better the text representation, the easier it is for the following phase to identify patterns for assigning the right class to a text. Thus, the text representation based on biLSTMs not only enables a more accurate text classification in the following phase but also offers a past and a future context view of the text, which allows to further enhance the text classifiers based on the unlabeled data through co-training.

### 3.3 Training Classifiers: Expanding Classifiers' Capabilities by Co-training

The aim of the second phase is to enhance the accuracy of machine learning classifiers for text classification by providing them with additional training data based on unlabeled data. To reach this goal, unlabeled texts are labeled in an automated manner. One of the most famous approaches exploiting the knowledge in unlabeled data by automated labeling is co-training [19]. Indeed, co-training approaches are applied with convincing results for text classification [3–6, 27]. Further on, co-training-based

approaches are designed to take advantage of two different views or representations of data as given by the context-dependent embeddings from the first phase. Consequently, we rely on a co-training-based approach to leverage text classifiers' capabilities based on auto labeled texts and hence, reduce time and effort for labeling.

Transferring the original co-training approach to the task of text classification it can be described as the process of training two classifiers and successively retrain both after providing each classifier with text labeled by the respective other classifier [19]. In detail, co-training requires two different views of text in order to train each classifier based on the set of labeled texts $L$ but each based on a different view of the texts. Subsequently, both classifiers are employed to classify or rather auto label a subset of unlabeled texts $U'$ randomly chosen of the set of unlabeled texts $U$. The $X^c$ most confidently auto labeled texts for each class are then added to the set of labeled texts $L$. In the case of machine learning classifiers, this confidence is determined by the probabilities assigned by the classifiers regarding the auto labeled texts. One iteration of co-training is closed by retraining the classifiers on the expanded set of labeled texts $L$. Accordingly, the next iteration starts by drawing a new subset $U'$ of the unlabeled texts $U$. This procedure is repeated until all texts of the set of unlabeled data $U$ have been labeled or a predefined number of iterations $k$ is reached. The concept behind co-training, which enhances both classifiers with each iteration, is based on the two different views used to train the classifiers. Since one classifier provides the other with labels it is most certain, these auto labels show a high probability of being correct while they may provide a higher degree of difficulty of classification for the classifier operating with the other view. By this means, both classifiers are provided with different auto labeled texts they are not necessarily certain how to classify by themselves and hence, can be improved by training on them without the need of labeling texts by hand.

In our case, the context-dependent embeddings from the first phase of our approach provide two different views by design. One view is represented by the part of the embeddings based on the past context words $(w_1, \dots, w_{i-1})$ whereas the other view is based on the future context words $(w_{i+1}, \dots, w_n)$. Consequently, we design our second phase, by splitting up the context-dependent embeddings so that co-training can be approached by training one classifier based on each part of the context-dependent embeddings. Further on, we decided to not limit the design of this phase to a specific co-training approach. Indeed, there exist several recent extensions of co-training, which might enhance the classical co-training approach depending on the text classification task and the dataset [3, 4]. For instance, Chen et al. [4] refined the selection of the auto labels, which are added to the set of labeled texts $L$ in each iteration. By employing their "double-check" strategy they only select those auto labels for training which are assigned by both classifiers to the same class and additionally provide a given similarity to manually labeled texts of the same class. Others, as Katz et al. [3], develop co-training further by saving in each iteration the respective classifiers and thereby train an ensemble of classifiers. Of course, implementing the mentioned approach requires greater memory capacities. Going a step further, also a combination of compatible co-training approaches, as those of Chen et al. [4] and Katz et al. [3], represents a conceivable realization of this phase.

In order to combine the classifiers after co-training for deployment in text classification tasks, we propose a merging function $m(\ p_1^c, \dots, p_n^c\ )$. By this means, the probabilities $p_i^c$ assigned from each classifier $i$ that a given text corresponds to class $c$ can be offset to a unified class probability. For example, in accordance with the original co-training approach, the average of the probabilities $p_i^c$ can be used to specify the function $m(\ p_1^c, \dots, p_n^c\ )$ [19].

To sum up, the second phase generates auto labels to expand the training data for text classifiers by relying on co-training. To do so, the texts represented by the context-dependent embeddings from the first phase are split into two views necessary for co-training based on the past and future context contributing to the embedding. As a result, co-training can be approached by training one classifier per view and using each for generating auto labels for the respective other classifier. Subsequently, the classifiers can be retrained based on the labeled and auto labeled texts. This procedure is repeated multiple times to stepwise increase the labeled data and hence, improve the classifiers' capabilities. Thus, the time and expenses to label texts by hand can be reduced. Finally, the trained classifiers can be employed for text classification tasks while their outputs are combined by a merging function $m(\ p_1^c, \dots, p_n^c\ )$.

# 4 Demonstration and Evaluation

## 4.1 Case Setting and Dataset

In order to demonstrate the practical applicability and evaluate the effectiveness of our approach, we used the case of a German direct banking institution. The institution is specialized in the field of community banking and maintains an online social network where customers are encouraged to discuss issues regarding financial services and products. For instance, users have the opportunity to discuss the conditions to obtain a loan from the bank or exchange experiences about saving and investment. In order to monitor the mood within public forums and be able to intervene when users continue to negatively impact the atmosphere, an adequate text classification approach is needed. In this case, texts shall be classified regarding their sentiment resulting in the classification task of sentiment analysis. In particular, the money forum, where concrete financial investment opportunities are shared or new financial products and services are proposed, reaches a high popularity amongst users of the banking institution. As a consequence, a domain-specific language is used within the money forum so that texts from the forum have to be labeled to be able to train a well-adapted classifier. For these reasons, the money forum provides an appropriate setting to apply our novel hybrid approach in order to reduce time and expenses associated with manually labeling texts for text classification approaches based on machine learning. Therefore, the banking institution provided us with a unique dataset comprising 308,087 texts written between the 1st September 2009 and 11th November 2016 in German language. The dataset contains on average approximately 31 words per text.

### 4.2 Demonstration of Our Approach for the German Bank

In the following, as an essential part of the Design Science research process (cf., e.g. Peffers et al., [21]), we demonstrate the applicability of our approach. To do so, a small amount of labeled texts is required. Hence, 3,000 randomly selected texts of our dataset have been labeled into the classes positive, negative and, as recommended by Go et al. [29], neutral. As a result, we obtained 612 (20.40%) texts labeled positive, 661 (22.03%) texts labeled negative and 1,727 texts (57.57%) belonging to the neutral class. We use 80% of these labeled texts as training data within our approach and keep the remaining 20% as test set for evaluation purposes in Subsection 4.3 while retaining the class distribution for each set.

Following the first phase of our approach, we used the recent and well-known ELMo approach to reach context-dependent embeddings trained via language modeling [7]. In this regard, we used the pre-trained *ELMoForManyLangs* python implementation from Che et al. [30] providing an ELMo model for the German language. By this means, we were able to represent both labeled and unlabeled texts as context-dependent embeddings. In detail, the used ELMo implementation provides the average of three different embeddings extracted from different consecutive layers of the ELMo model (cf. Figure 2).

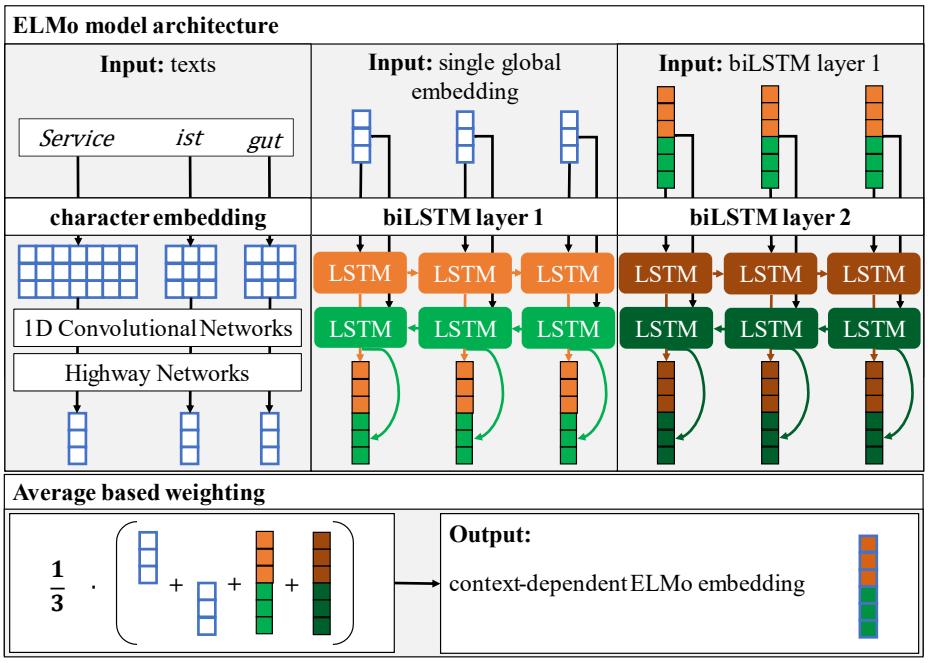


**Figure 2.** Context-dependent ELMo embeddings as implemented by Che et al. [30]

On the one hand, a single global embedding arising from the output of a so-called character level Convolutional Neural Network in terms of a vector with 512 units. This type of embedding is gained by processing each word character by character, thereby

representing each character as an embedding vector and further processing all character embeddings from each word with, at first, 1D-Convolutional Networks and subsequently with Highway Networks to reach a single global embedding for each word. For further details of this architecture, we refer to Kim et al. [20]. On the other hand, two context-dependent representations are gained from two consecutive biLSTM layers, each producing a vector with 1,024 units per word based on 512 units from the past context and 512 units from the future context, respectively. The average of the three embeddings is computed by first, building a new vector with 1,024 units using the 512 units of the single global embedding twice (once for the past and once for the future context) and second, computing the average of the three vectors unit-wise. As a result, each word of labeled and unlabeled texts is represented by an ELMo vector comprising 1,024 units of averaged embeddings.

For the second phase of our approach, an implementation of a co-training approach is required to label texts in an automated manner. In this regard, we rely on the original co-training approach [19] so that implementations of enhanced co-training approaches could even further improve the results. To do so, we used two Multi-Layer Perceptrons (MLPs) with the same architecture trained by the Keras module within the Tensorflow 2.0 library [23]. We provide each MLP with a different view on the texts by splitting the context-dependent ELMo embeddings from the first phase based on the past and future context units. By this means one MLP is trained by using the first 512 units of each ELMo embedding while the other is trained based on the last 512 units per word. To find a sound architecture and configuration for our MLPs as well as for the parameters of our co-training implementation, we had limited access to an NVIDIA Tesla P100 GPU via the Google Cloud Platform[1]. As a result, the architecture of each MLP comprises an input, three dense and an output layer. The input layer receives the concatenated embeddings from the first 50 words of each text. Since MLPs require inputs of the same size and the average text contains around 31 words, we found that a padding to 50 words is a sound configuration to avoid large sparse vectors for shorter texts while at the same time being able to adequately process longer texts. The dense layers contain from bottom to top 128, 64 and 32 neurons all using ReLU as activation function. The output layer consists of three neurons, each for one class, activated by the softmax function. We chose only 10% of the training data as validation data to still have enough labeled texts for actual training. On this basis, we trained our MLPs for 50 epochs with a batch size of 64 using early stopping if the loss on the validation data does not decrease within three epochs. As usual for classification tasks, we used the categorical cross entropy as loss function and employed the adam optimizer. To parameterize our co-training implementation, we found that a subset of $U' = 1{,}000$ randomly selected unlabeled texts per iteration constitutes a sensible choice. Further, we generate our auto labels within $k = 40$ iterations of our co-training, while expanding the set of labeled texts $L$ in each iteration by the respective most confidently labeled 11 negative, 27 neutral and 10 positive texts. By this means, we retain the class distribution similar to Blum and Mitchell [19]. In line with research, we specify our merging function $m(\, p_1^c, p_2^c \,)$ as the average of the probabilities $p_i^c$ per class $c$ assigned

[1] https://cloud.google.com/

by the MLPs [3, 19]. By this means, each classifier contributes to the same extent to the final classification result.

Summing up, to train two MLPs by exploiting unlabeled texts, we relied on pre-trained context-dependent embeddings from the ELMo approach and implemented the original co-training approach to generate auto labels. Hence, institutions can be provided with a sound text classification while the time and expenses associated with manual labeling of texts can be reduced.

### 4.3 Evaluation

In order to evaluate our approach, we compared its performance on the test set against that of well-established competing artifacts for text classification [7, 23, 31]. To ensure comparability, all considered approaches are based on the same MLP architecture introduced within the demonstration of our approach in Section 4.2 while varying the input layer based on the competing text representation. In this regard, we chose three approaches as baselines for comparison. First, the most widely used statistical term weighting representation tf-idf [31]. Thereby, each word is weighted based on the frequency of its occurrence in the respective text as well as in the whole dataset. More precisely, each word occurrence in the respective text to be classified increases its tf-idf weighting and hence, increases influence on the classification output. In turn, the weighting for frequently occurring words in the whole dataset decreases. This is due to the idea that a frequently occurring word does not add specific value to the text to be classified. The representation of text by the tf-idf weighting can be improved by an adequate pre-processing. Thus, we applied pre-processing insofar as words had been cleared from stop words, transformed to lower case and reduced to their word stems. Second, a single global embedding representation learned via using an embedding layer as input layer [23]. By this means, single global embeddings are gained simultaneously when training the MLP. Third, the context-dependent embeddings from the ELMo approach using the pre-trained German embeddings from the *ELMoForManyLangs* implementation [30]. Further on, we report the results of our hybrid approach for multiple numbers of co-training iterations $k$. Please note that a comparison to a competing co-training approach using two different text representations (e.g. ELMo and CoVe) as views is not fair, since further information would be added. Actually, such a broader co-training approach could be improved by using our hybrid approach in each view leading to a recursive co-training approach.

To assess text classification performances, we calculated the well-known metrics *accuracy* as well as the $F_1$*-Score* for each class. Since our test set is rather small and text classification performance of MLPs can vary based on the randomly chosen initial weights, we repeated the training of the classifiers for 40 times and thereby determined the macro-average for our evaluation metrics. Additionally, we report a 99%-confidence interval for the macro-averages, relying on the *t*-distribution, which is often applied for building confidence intervals for a mean based on a small sample size. On this basis, we were able to rigorously evaluate text classification approaches in our setting (cf. Table 1).

**Table 1.** Evaluation of our approach in comparison with competing artifacts

| **Competing artifacts /Metrics in percent** | **Accuracy** | **$F_1$-Score** (positive) | **$F_1$-Score** (neutral) | **$F_1$-Score** (negative) |
|---|---|---|---|---|
| **Tf-idf** | 56.38 ± 0.96 | 22.82 ± 5.75 | 71.30 ± 0.92 | 26.4 ± 3.84 |
| **Global embedding** | 57.75 ± 0.27 | 0.45 ± 1.11 | 73.09 ± 0.18 | 9.17 ± 5.01 |
| **ELMo** | 61.13 ± 0.70 | 29.05 ± 4.72 | 75.49 ± 0.59 | 35.07 ± 4.49 |
| **Hybrid approach** ($k$=0) | 61.30 ± 0.58 | 21.02 ± 4.69 | 75.57 ± 0.50 | 33.35 ± 3.03 |
| **Hybrid approach** ($k$=10) | 62.50 ± 0.60 | 31.12 ± 3.80 | 76.50 ± 0.51 | 36.50 ± 3.41 |
| **Hybrid approach** ($k$=20) | 63.05 ± 0.44 | 33.25 ± 2.60 | 76.80 ± 0.46 | 38.42 ± 2.61 |
| **Hybrid approach** ($k$=40) | **63.18** ± 0.52 | **34.16** ± 3.57 | **76.82** ± 0.46 | **39.04** ± 2.18 |

Accordingly, our hybrid approach started with an accuracy of 61.30% (±0.58%) without any auto labels from co-training ($k$=0) and gradually increased with co-training iterations. In contrast, the ELMo approach reached an accuracy of 61.13% (±0.70%). In this regard, it was to be expected that the ELMo approach does only marginally differ from the initial state of our approach for $k$=0 as both receive the same information and differ only in the processing. While the ELMo approach receives the full context-dependent embeddings and processes them by one MLP classifier, our approach for $k$=0 receives the splitted ELMo embeddings, processes them by two MLP classifiers and outputs the merged results. However, already $k$=10 iterations of co-training within our approach are enough to outperform the competing approaches for all of the evaluation metrics. Additionally, further iterations further improve results for each metric. Moreover, after $k$=10 iterations even the confidence intervals for the accuracy of our approach do not touch those of the competing approaches. With 57.75% (±0.27%) the directly trained single global embeddings had the second lowest value for accuracy. However, as can be seen from the $F_1$-Scores, this approach assigned most of the texts to the neutral class. Although the tf-idf baseline performed worst in terms of accuracy, it was able to distinguish the three classes to some extent as reflected by the $F_1$-Scores for each class. Indeed, our results are in line with literature as accuracy for sentiment analysis for short messages is often below 60% for the multiclass case including a neutral class [32].

To gain more detailed insights with respect to reducing manual labeling of texts when applying our approach, we evaluated our approach using only 2,000 instead of 2,400 labeled texts for training our MLP classifiers. On this basis, our approach started with an initial accuracy of 60.13% (±0.63%) for $k$=0 iterations while also obtaining $F_1$-Scores below the ELMo approach. Surprisingly, we reached similar results for $k$=10 as those in Table 1 when all labeled texts have been used for training. For $k$=40 iterations

of our approach with reduced labeled texts we obtained an accuracy of 62.70% (±0.63%) and $F_1$-Scores for the positive, neutral and negative class of 32.24% (±2.58%), 76.23% (±0.43%) and 40.38% (±2.40%). Hence, our approach is even able to outperform the competing artifacts when trained with less labeled texts.

## 5 Conclusion, Limitations and Future Research

Machine learning is nowadays becoming more and more established in companies while offering great potential for the task of text classification as sophisticated patterns have to be identified. However, training sound text classification approaches via machine learning requires a decent amount of manually labeled texts. Since labeling texts requires time and is associated with expenses, research provides promising approaches to exploit the knowledge in unlabeled texts by learning sophisticated semantic representations [7, 8, 14–18, 20] or by labeling texts in an automated manner through co-training [3–6, 19]. Nevertheless, until now literature does not provide sufficient approaches combining these two research streams. Hence, we contribute to research and practice by proposing a novel hybrid text classification approach combining text representations based on biLSTMs with co-training approaches in an integrated perspective to reduce time and expenses associated with labeling texts. Our approach takes benefit of the past and future context representations obtained from biLSTMs by integrating them as different views into co-training. We demonstrated and evaluated our approach using the case of a German bank. The results of the evaluation reveal that our hybrid approach provides greater text classification capabilities compared to other state-of-the-art approaches even if trained with less labeled texts.

Nevertheless, our work also has some limitations which may constitute the starting point for future research. In this paper we focused on embeddings based on biLSTMs from an in-depth perspective. Future research could further exploit unlabeled texts by adding different single global embeddings (e.g. Word2Vec and GloVe) to the respective views in our approach or even design recursive co-training approaches by integrating our approach as one view in a broader co-training approach. Furthermore, we only considered one dataset, for which we applied and evaluated our approach. As in our case the dataset was skewed towards the neutral class, it would be interesting to investigate the performance of our approach on datasets with different class distributions (e.g. equally sized classes) as well as with further variations regarding the amount of labeled texts. Summing up, we believe that our hybrid approach is an important step towards combining embeddings based on biLSTMs with co-training. Going a step further, the question arises how to ensure that co-training approaches do not learn undesired patterns when trained on auto labeled texts. As promising starting point, it seems reasonable to make use of explainable artificial intelligence approaches to retrace results and guarantee plausibility [33]. With this in mind, we hope to stimulate future research to push this exiting research field forward.

## References


1. International Data Group Research Services: Studie Machine Learning 2020 (2020)
2. Algorithmia: 2020 state of enterprise machine learning (2020)
3. Katz, G., Caragea, C., Shabtai, A.: Vertical Ensemble Co-Training for Text Classification. ACM Transactions on Intelligent Systems and Technology 9, 1–23 (2017)
4. Chen, J., Feng, J., Sun, X., et al.: Co-Training Semi-Supervised Deep Learning for Sentiment Classification of MOOC Forum Posts. Symmetry 12, 8 (2019)
5. Karisani, P., Ho, J., Agichtein, E.: Domain-Guided Task Decomposition with Self-Training for Detecting Personal Events in Social Media. In: Web Conference, pp. 2411–2420 (2020)
6. Kim, D., Seo, D., Cho, S., et al.: Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec. Information Sciences 477, 15–29 (2019)
7. Peters, M., Neumann, M., Iyyer, M., et al.: Deep Contextualized Word Representations. In: NAACL, pp. 2227–2237. ACL (2018)
8. McCann, B., Bradbury, J., Xiong, C., et al.: Learned in translation: Contextualized word vectors. In: NIPS, pp. 6294–6305 (2017)
9. Liu, Q., Kusner, M.J., Blunsom, P.: A Survey on Contextual Embeddings. ArXiv (2020)
10. Qiu, X., Sun, T., Xu, Y., et al.: Pre-trained Models for Natural Language Processing: A Survey. ArXiv (2020)
11. Bitkom: Big Data und Geschäftsmodell-Innovation in der Praxis: 40+ Beispiele (2015)
12. Ratner, A., Bach, S.H., Ehrenberg, H., et al.: Snorkel: Rapid training data creation with weak supervision. In: International Conference on VLDB, 11, pp. 269–282 (2017)
13. Cognilytica Research: Data Engineering, Preparation, and Labeling for AI 2020. Getting Data ready for Use in AI and Machine Learning Projects (2020)
14. Mikolov, T., Chen, K., Corrado, G., et al.: Efficient estimation of word representations in vector space. ArXiv (2013)
15. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Conference on EMNLP, pp. 1532–1543 (2014)
16. Devlin, J., Chang, M.-W., Lee, K., et al.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL, pp. 4171–4186. ACL (2019)
17. Kawakami, K., Dyer, C.: Learning to Represent Words in Context with Multilingual Supervision. In: Workshop ICLR (2016)
18. Melamud, O., Goldberger, J., Dagan, I.: context2vec: Learning generic context embedding with bidirectional lstm. In: 20th SIGNLL CoNLL, pp. 51–61 (2016)
19. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: 11th COLT, pp. 92–100 (1998)
20. Kim, Y., Jernite, Y., Sontag, D., et al.: Character-aware neural language models. ArXiv (2015)
21. Peffers, K., Tuunanen, T., Rothenberger, M.A., et al.: A Design Science Research Methodology for Information Systems Research. JMIS 24(3), 45–77 (2007)
22. van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Machine Learning 109, 373–440 (2020)
23. Abadi, M., Barham, P., Chen, J., et al.: Tensorflow: A system for large-scale machine learning. In: 12th USENIX OSDI, pp. 265–283 (2016)
24. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. ArXiv (2018)
25. Peters, M.E., Ammar, W., Bhagavatula, C., et al.: Semi-supervised sequence tagging with bidirectional language models. ArXiv (2017)

26. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks 18, 602–610 (2005)
27. Wu, J., Li, L., Wang, W.Y.: Reinforced co-training. ArXiv (2018)
28. Lim, K., Lee, J.Y., Carbonell, J., et al.: Semi-Supervised Learning on Meta Structure: Multi-Task Tagging and Parsing in Low-Resource Scenarios. In: AAAI Conference (2020)
29. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N project report, Stanford 1 (2009)
30. Che, W., Liu, Y., Wang, Y., et al.: Towards Better UD Parsing: Deep Contextualized Word Embeddings, Ensemble, and Treebank Concatenation. In: CoNLL 2018, pp. 55–64 (2018)
31. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information processing & management 24, 513–523 (1988)
32. Socher, R., Perelygin, A., Wu, J., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Conference on EMNLP, pp. 1631–1642 (2013)
33. Holzinger, A., Kieseberg, P., Weippl, E., et al.: Current advances, trends and challenges of machine learning and knowledge extraction: from machine learning to explainable AI. In: International CD-MAKE, pp. 1–8 (2018)